DOCUMENT RESUME

ED 084 878 EM 011 702

AUTHOR	Mitchell, Ron; Conner, Michael
TITLE	User's Guide to Coursewriter II Preprocessor, VI. 2. Systems Memo Number Three.
INSTITUTION	Texas Univ., Austin. Computer-Assisted Instruction Lab.
SPONS AGENCY	National Science Foundation, Washington, D.C.
REPORT NO	SM-3
PUB DATE	May 71
NOTE	59p.; See also EM 011 700 and EM 011 701
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Programs; *Computer Science; Manuals; *Programers; *Programing
IDENTIFIERS	*Coursewriter II Preprocessor

ABSTRACT

Procedures are presented to assist programers with the use of the Coursewriter II preprocessor. The preprocessor was designed to reduce the repetition of effort which takes place from the conception of a course by its author to the time of its use by students in on-line instruction. It accomplishes this by relieving the programer of the tedious task of translating planning guides into Coursewriter II statements, allowing him to concentrate upon program logic. Topics covered in this manual include code descriptions, variable reference, error messages and recovery, coding conventions, job submission, output, course assembly, and malfunctions. A sample problem and a glossary of terms are appended. (PB)

ED 084873

THE UNIVERSITY OF TEXAS AT AUSTIN

Computer Assisted Instruction Laboratory

AUSTIN

EM 011 104

FILMED FROM BEST AVAILABLE COPY

ED 084878

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

USER'S GUIDE

to

COURSEWRITER II PREPROCESSOR, V1.2

SYSTEMS MEMO NO. 3

Ron Mitchell
Michael Conner

May 1971

Supported By:

THE NATIONAL SCIENCE FOUNDATION
Grant GJ 509 X

Computer-Assisted Instruction Laboratory
C. Victor Bunderson, Director
The University of Texas at Austin
Austin, Texas 78712

TABLE OF CONTENTS

INTRODUCTION 1

OP-CODE DESCRIPTIONS
- GENERAL 5
- BEGIN PROBLEM(BP) 6
- VARIABLE DECLARATION(VD) 7
- MESSAGE DEFINITION(MD) 9
- DISPLAY MESSAGE(DM) 11
- PUNCH TABLE(PT) 12
- END PROBLEM(ZZ) 14

VARIABLE REFERENCE 15

ERROR MESSAGES AND RECOVERY 16

CODING CONVENTIONS
- 6600 INPUT CONVENTIONS 20
- SUGGESTED CODING CONVENTIONS 21

6600 JOB SUBMISSION 22

PREPROCESSOR PRINTED OUTPUT 23

PREPROCESSOR PUNCH OUTPUT 25

PROCESSING PUNCH OUTPUT 26

COURSE ASSEMBLY 29

UPDATES AND PREPROCESSOR MALFUNCTIONS 31

APPENDIX A: KNOWN MALFUNCTIONS

APPENDIX B: SAMPLE PROBLEM
- PLANNING GUIDE
- PREPROCESSOR INPUT LISTING
- PREPROCESSOR OUTPUT LISTING
- ASSEMBLED PROBLEM LISTING

GLOSSARY OF TERMS

This document was processed for the ERIC Document Reproduction Service by ERIC/EM. Some pages may not be readable in microfiche or hard copy. However, this is the best available copy, and we feel that the document should not be withheld from interested readers on the basis ofthese unreadable pages alone.

The preprocessor was designed to reduce the repetition of effort which takes place from the time of conception of a course by its author to the time of having it available for on-line student instruction. The process of implementation is normally stepwise as shown:

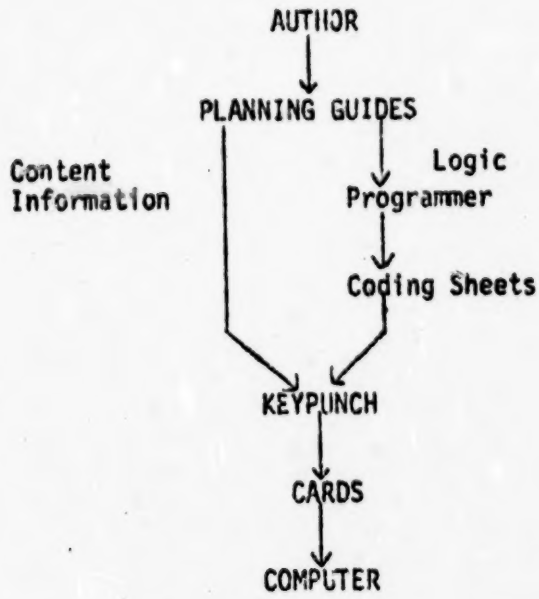
AUTHOR
↓
PLANNING GUIDES
↓
PROGRAMMER
↓
CODING SHEETS
↓
KEYPUNCH
↓
CARDS
↓
COMPUTER

The programmer deals mainly with two types of information: control logic or course structure information and content information. The preprocessor was designed to enable the programmer to deal with the latter of these in a more natural way than CW II allows.

In most cases the course content information is completely and explicitly stated in the author's planning guides and the task of the programmer is, for the most part, that of the tedious and completely mechanical translation of the planning guides into the appropriate sequence of CW II statements. It is difficult, if not impossible, to perform this tedious translation on even small amounts of material, much less the enormous amounts required by a major course, without introducing a tremendous number

of errors.

The solution to this problem is to make use of that content information which is explicitly stated in the planning guides and thus relieve the programmer as much as possible of this mechanical task and enable him to concentrate on the program logic. The steps in going from course conception to implementation would be:



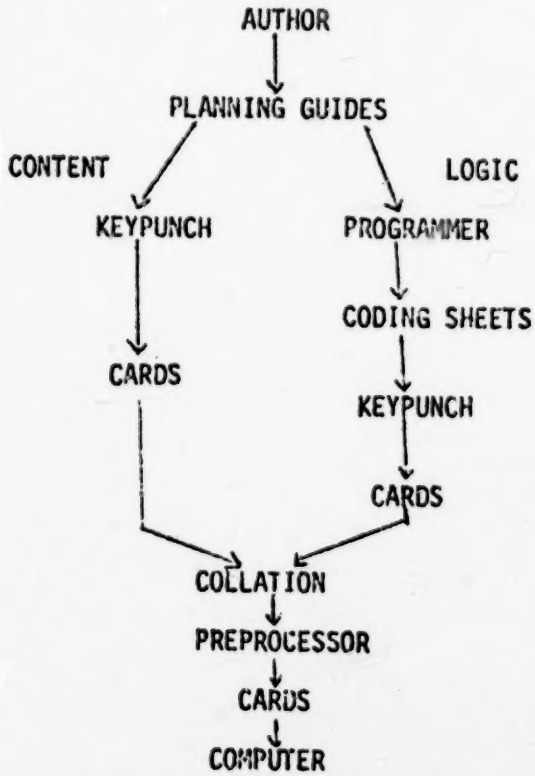
In making this bypass, the following constraints must be observed:

1. Keypunching must remain more or less a transcription task,
2. No additional restrictions must be placed on the author.
3. A means for collation of the transcribed course content information and the programmer-produced course logic must be provided.
4. The programming of the logic should be made easier and clearer by allowing the programmer a convenient and natural (programming-wise) means of referencing the course content.

The preprocessor was designed to make this bypass according to the above criteria.

The preprocessor has added six op-codes to the CW II language. These op-codes allow course content information to be entered directly from the planning guides by the keypunch operator and referenced within the flow of course logic by the programmer.

The content information which is explicitly stated in the planning guides and recognized by the preprocessor is of two types: message and variable. Message content data are display templates which are constant except for references to variable data. Variable data consists of variable length literal strings and references to these variables may be made in any way that a literal string (in a buffer) is referred to by its buffer specification in CW II. With the preprocessor the flow from author to computer would be as shown in the following diagram.



The keypunch operator can define message content and variable data from the planning guides (with a minimal amount of effort by the programmer) by use of certain of the op-codes and the programmer has easy access to these data by variable reference and by use of another of the op-codes. Collation has been reduced to a trivial task since the information transcribed by the keypunch operator is analogous to declarations in ALGOL and appears before any program logic produced by the programmer.

This user's guide assumes a knowledge of CW II.

OP-CODE DESCRIPTIONS

The following table describes the six op-codes which have been added to CW II by the preprocessor. Brief examples are included with each table entry and notes on use are provided as need is seen. The section on known malfunctions should be read to avoid situations which will cause these to occur.

Unless otherwise specified in the table, all entries follow the normal CW II instruction format, i.e., op-codes begin in column 7 with columns 9-10 blank(except for EOB in column 9).

The examples shown in the table are from preprocessor output.

BEGIN PROBLEM:

BP P_1⊗/P_2≠

Standard dictionary number for course of which this problem is a part. Default value is zero(system dictionary).

Three-letter base label. All labels generated by the preprocessor for the problem have this label as base,i.e., first three characters. Default value is AAA.

This op-code signifies the beginning of a problem and defines the base-label* for the problem and also indicates the standard dictionary for the course of which the problem is a part.

EXAMPLE:

BP FAC⊸/1≠

*See glossary of terms.

This op-code allows variable data* to be declared and variable attributes assigned(as shown on the preceding page) and causes a table of variable values consisting of the definition-sets* of the data-set* to be created. A maximum of 100 variables may be declared and the data-set must contain no more than 24K characters including EOS's delimiting definition-sets and alternate-coded slashes delimiting variable values within definition-sets. (See the section on known malfunctions before using this op-code.)

EXAMPLE:

```
      VD  1⍎
          N1¨/13¨/15¨/1⍎
          N2¨/19¨/20¨/1⍎
          N3¨/2¨/5⍎
          C1¨/3¨/5⍎
          C3¨/3¨/5⍎
          N4¨/13¨/15¨/1⍎
          C4¨/3¨/5⍎
⍎
1¨/0 <2⍟>≥ X¨/X <2 >0¨/0¨/YES¨/NO¨/0 <2⍟>≥ 0¨/YES⍎
2¨/0 <2⍟>≥ X¨/X = 3¨/3¨/YES¨/YES¨/0 <2⍟>≥ 3¨/YES⍎
3¨/0 <2⍟>≥ X¨/1 <2⍟>≥ X¨/-1¨/NO¨/YES¨/0 <2⍟>≥ -1¨/YES⍎
4¨/X =⍟/ X¨/X <2 >0¨/-1¨/NO¨/YES¨/-1 =⍟/ -1¨/YES⍎
5¨/1 <2⍟>≥ X <2⍟>≥ 10¨/12 <2 >X <2⍟>≥ 20¨/3¨/YES¨/NO¨/1 <2⍟>≥ 3 <2⍟>≥ 1X
0¨/YES⍎
6¨/0 <2⍟>≥ X¨/0 <2 >X AND X IS EVEN¨/2¨/YES¨/YES¨/0 <2⍟>≥ 2¨/YES⍎
7¨/X IS RATIONAL¨/X IS IRRATIONAL¨/¨P¨/NO¨/YES¨/¨P IS RATIONAL¨/YES⍎
8¨/X⍟2⍎- X = 12¨/X⍟2⍎<2 >9¨/4¨/YES¨/NO¨/4⍟2⍎- 4 = 12¨/YES⍎
9¨/0 <2⍟>≥ X¨/X <2 >0¨/0¨/YES¨/NO¨/0 <2⍟>≥ 0¨/YES⍎
10¨/0 <2⍟>≥ X¨/X = 3¨/3¨/YES¨/YES¨/0 <2⍟>≥ 3¨/YES⍎
11¨/0 <2⍟>≥ X¨/1 <2⍟>≥ X¨/2¨/YES¨/YES¨/0 <2⍟>≥ 2¨/YES⍎
12¨/X =⍟/ X¨/X <2 >0¨/4¨/NO¨/NO¨/4 =⍟/ 4¨/NO⍎
13¨/1 <2⍟>≥ X <2⍟>≥ 10¨/12 <2 >X <2⍟>≥ 20¨/0¨/NO¨/NO¨/1 <2⍟>≥ 0 <2⍟>≥ X
10¨/NO⍎
14¨/X⍟2⍎- X = 12¨/0 <2 >X AND X IS EVEN¨/5¨/NO¨/NO¨/5⍟2⍎- 5 = 12¨/NO⍎
15¨/<,>X<, >= 3¨/<,>X - 3<, >= 0¨/4¨/NO¨/NO¨/<,>4<, >= 4¨/NO⍎
16¨/<,>X<, 2 >3¨/5 <2 >X¨/1¨/YES¨/NO¨/<,>1<, 2 >3¨/YES⍎
⍎
```

*See glossary of terms.

MESSAGE DEFINITION:

MD≠

Following the MD op-code is a series of message definitions. Each definition consists of a message or display name(beginning in column 1) and a message body(beginning in column 1) and takes the following form:

[MESSAGE NAME]≠ — The name may be any legal CW II label.

[ROW NO.]0/[TEXT]≠

- [TEXT] — Any string of allowed CW II characters(see Coding Conventions) except EOB.
- [ROW NO.] — Beginning row number of the display. Must be respecified after an EOB has terminated a text string.

The last entry in a series of message definitions <u>must be</u> an EOB in column 1.

This op-code allows message content data*(pages of a planning guide) to be defined and assigned labels by which they may later be referenced. The message data may in turn contain references to variable data.

The only real restrictions which must be observed involve displays which include inserts, i.e., those which would include <u>dti</u>'s if coded conventionally. The restriction is that only one insert may be made for a given rectangular screen area and that insert must either begin in column 0 or end in column 39. The preprocessor recognizes that a <u>dti</u> must be generated when an ampersand(&) is encountered within the text portion of a message body*. The ampersand acts as a "don't-erase" character so that all positions covered by a string of ampersands are left when the <u>dti</u> is generated. An example is given on the following page.

*See glossary of terms.

EXAMPLE:

```
        ND#
FAC1#
0¬/<FAC≡↓R>ECALL FROM <SET NOTATION >THAT IF <L>(X) IS≡↓A STATEMENT, THX
EN Y ¬E ¬CX<: L>(X)¬V IF AND≡↓ONLY IF <L>(Y) IS TRUE.#
9¬/<S>UPPOSE  <P>(X) = :N1:≡↓↓          <Q>(X) = :N2:#
16¬/<I>S  <P>(:N3: ) TRUE</#
FAC2#
19¬/                    <G>OOD#
FAC3#
19¬/<S>ORRY, THAT'S FALSE.#
23¬/<P>(:N3: ) STANDS FOR :N4:#
27¬/<N>OW IS THIS TRUE OR NOT</  T>HE ANSWER≡↓IS :C1:  .#
FAC4#
19¬/<I>S  :N3: IN ¬CX<: P>(X)¬V</#
FAC5#
22¬/                        <C>ORRECT#
FAC6#
22¬/<N>OW LOOK AT THE PREVIOUS ANSWER AND RE¬≡↓MEMBER THAT :N3:  IS IN X
THAT SET IN CASE≡↓<P>(:N3: ) IS TRUE.#
29¬/<T>HE ANSWER IS :C1:  .#
FAC7#
23¬/<I>S  :N3:  IN ¬CX<: Q>(X)¬V</#
FAC8#
26¬/                        <G>OOD#
FAC9#
27¬/<F>ALSE.  <I>S <Q>(:N3: ) TRUE</≡↓↓T>HE ANSWER IS :C3:  .#
FAC10#
15¬/<I>S THE STATEMENT↓↓↓↓↓↓↓↓↓↓↓↓<P>(:N3: ) OR <Q>(:N3: )≡↓↓TRUE</#
FAC11#
21¬/^^^^^^^^^^^^^^^^^^^^^^^^<E>XCELLENT#
#
```

Message FAC11 is an example of an insert beginning in column 21.
Message FAC10 is an example of message content data which contains variable references.

DISPLAY MESSAGE:

DM P_1#

Name of the message to be displayed.

This op-code enables the programmer to reference, by label, previously defined message content data or, in different notation, the series of CM II statements generated to display such message content data.

EXAMPLE(including generated statements):

```
DM   FAC10#
LD   <I>S THE STATEMENT++++++++++<P>() OR <O>()≡++TRUE<¬/¬/>R4#
FN   MV¬/R2,35.05¬/R4,34¬/IA#
FN   MV¬/R2,35.05¬/R4,27¬/IA#
FN   ED¬/R4¬/¬/¬/A#
DT   15.00¬/0R,15¬/40.00¬/R4#
DM   FAC11#
DTI  21.00¬/02.21¬/20.20¬/XXXXXXXXXXXXXXXXXXXXX<E>XCELLENT#
```

PUNCH TABLE:

PT P_1,P_2,P_3

- P_3 — Iteration counter to be used in determining the definition number of the problem variables. Default value is C1.
- P_2 — Label of return point after buffer load.
- P_1 — Table(data-set) number, 0-9.

This op-code causes generation of a series of statements that provide the means by which variable data* take on new values. In certain instructional sequences, the displays may be constant except for questions presented the student(and associated non-displayed answers) and feedback. A method which allows the same block of code to be used for several presentations (iterations) of a problem with different variable data each iteration is to branch to a sequence of code which loads the variable data into buffer(s) and then return to the beginning of the segment for problem presentation. The variable data are extracted from the buffer(s) for display. The statements generated provide for selection of the definition-set* (depending on the value of the iteration counter) from which values are assigned to variables, assignment of the values(buffer loads), and return branches to the sequence of code for presentation. These statements as a group may be accessed by branching to a label of the form XXXY00, where XXX is the base label* of the problem and Y is the data-set* number. The preprocessor also generates labels of the form XXXY followed by two digits(decimal). One label is generated for each definition-set* in the data-set* and the numeric portion of the label is the definition-set number(with leading zero if necessary); thus the programmer must not create labels of this form when using the PT op-code. Buffers 2 and 3 are used for loading the definition-set and are therefore considered to be dedicated if this op-code is used.(Buffer 3 is available for use if an entire definition-set contains less than 100 characters.) If the length of a variable in a given definition-set is less than the maximum buffer length specified, the remaining positions in the buffer are filled out with ampersands(&).

*See glossary of terms.

EXAMPLE(including generated statements):

```
        PT  1¬/FACA¬/C1#
FAC100#
        BR  FAC101¬/C1¬/E¬/01#
        BR  FAC102¬/C1¬/E¬/02#
        BR  FAC103¬/C1¬/E¬/03#
        BR  FAC104¬/C1¬/E¬/04#
        BR  FAC105¬/C1¬/E¬/05#
        BR  FAC106¬/C1¬/E¬/06#
        BR  FAC107¬/C1¬/E¬/07#
        LD  1¬/C1#
FAC101#
        LD  0 <2⍟>≥ X^^^^^^^^^X <2 >0^^^^^^^^^^^^^^^^0^^^^YES^^NO^^^0 <2⍟>≥X
 0^^^^^^^^^YES^^¬/B2#
        BR  FACA#
FAC102#
        LD  0 <2⍟>≥ X^^^^^^^^^X = 3^^^^^^^^^^^^^^^^3^^^^YES^^YES^^0 <2⍟>≥ 3X
^^^^^^^^^YES^^¬/B2#
        BR  FACA#
FAC103#
        LD  0 <2⍟>≥ X^^^^^^^^^1 <2⍟>≥ X^^^^^^^^^^^^^^-1^^^NO^^^YES^^0 <2⍟>≥X
 -1^^^^^^^^YES^^¬/B2#
        BR  FACA#
FAC104#
        LD  X =⍟/ X^^^^^^^^^X <2 >0^^^^^^^^^^^^^^^^-1^^^NO^^^YES^^-1 =⍟/ -1X
^^^^^^^YES^^¬/B2#
        BR  FACA#
FAC105#
        LD  1 <2⍟>≥ X <2⍟>≥ 10^12 <2 >X <2⍟>≥ 20^^^^^^^3^^^^YES^^NO^^^1 <X
2⍟>≥ 3 <2⍟>≥ 10^YES^^¬/B2#
        BR  FACA#
FAC106#
        LD  0 <2⍟>≥ X^^^^^^^^^0 <2 >X AND X IS EVEN^2^^^^YES^^YES^^0 <2⍟>≥X
 2^^^^^^^^^YES^^¬/B2#
        BR  FACA#
FAC107#
        LD  X IS RATIONAL^^X IS IRRATIONAL^^^^^¬P^^^^NO^^^YES^^¬P IS RATIX
ONAL^^YES^^¬/B2#
        BR  FACA#
```

END PROBLEM:

ZZ≠

This op-code signals the end of a problem and causes reinitialization of all pointers, flags, and tables.

EXAMPLE:

ZZ≠

VARIABLE REFERENCE

Variables may be referenced in preprocessor code in any manner that a literal string (in a buffer) is referred to by its buffer specification in CW II. A variable is referenced by delimiting the variable name with colons.

EXAMPLES:

```
FN   MV0/:C1:0/B40/A≠   (Valid)
LD   :C1:0/B4≠          (invalid)
CA   :C1:≠              (Invalid)
```

Variables may also be referenced during message definition (see example in table of op-codes) in any manner that a literal string is referred to in CW II <u>except</u> by referral to a buffer specification.

ERROR CODES

The first 10 columns of a line in preprocessor output are used as a flag field for messages to the programmer. When an error is encountered during preprocessing, an error message is output immediately following the erroneous statement. The message consists of the error number in the flag field, the statement identifier of the last correct statement processed, and the number of the preprocessor routine in which the error was detected. In some instances the error number is repeated on the following line along with an echo-print of the erroneous statement.

```
EXAMPLES: NO PUNCH     BAF00110          DM  BAFA≠
          ERROR   20   BAF00110  FOUND BY ROUTINE NUMBER  11
```

The following table gives the error numbers with probable cause and any recovery procedures.

ERROR NUMBER	CAUSE AND RECOVERY
1	No ;CB before column 72 and column 72 contains no continuation code. Next statement read.
2	Statement more than 250 characters long. Next statement read.
3	Statement after variable substitution is longer than 250 characters. Next statement read.
4	Value not specified for necessary parameter. Parameter list evaluation is terminated and next statement read.
5	Format error in parameter list, e.g., actual type differs from expected type, non-numeric character embedded in numeric parameter, alphabetic parameter longer than six characters. Next statement read.
6	Variable referenced which was not declared. Variable delimiters are set to X's and statement processing continues.
7	Format error in variable reference, e.g., right delimiter of reference omitted. Next statement read.
8	Illegal data-set(table) number. Next statement read.
9	More than 100 variables declared. Next statement read.
10	Variable definitions have overflowed data-table(more than 24K characters). Next statement read.
11	Definition-set will not fit into two buffers. Next statement read.

ERROR NUMBER	CAUSE AND RECOVERY
12	Doubly-defined data-set number. Next statement read.
13	Non-sequential definition-set number. Processing continues with definition-set number taking the non-sequential value.
14	Definition-set number greater than 99. Processing continues with definition-set number incremented by one over previous value.
15	Illegal row number specified. Next statement read.
16	Entire rectangular portion of display cannot be defined in 200 characters(99 if variable reference present). That portion of display is not generated and processing continues. Next statement read.
17	No text appears in the text portion of a message body. Next statement read.
19	Illegal message label(longer than 6 characters). Next statement read.
20	Referenced message was not defined. Next statement read.
25	More variables defined in in definition-set than were declared. Buffer load statement(s) generated and processing continues with next definition-set in data-set.
26	All variables declared were not defined in the definition-set. Buffer load statement(s) generated and next statement read.
27	Variable definition longer than maximum buffer length attributed. Definition is set to all X's and processing of definition-set continues. TRUNCATED appears in flag field of load statement generated.

ERROR NUMBER	CAUSE AND RECOVERY
28	Data-set referenced by PT not defined. Next statement read.
29	Data-set number outside the range 0-9. Next statement read.

CDC 6600 INPUT CODING CONVENTIONS

There are a few restrictions which must be observed concerning characters which may not appear in the input stream except for reserved uses.

A colon may not appear in the input stream except when used as a variable delimiter; however, it may be coded as an upshift semicolon. Automatic conversion to a colon is provided by the processor before punch output is produced.

Another restriction is limited use of the ampersand(&). The ampersand may be used freely <u>except</u> within message definitions. When the preprocessor encounters an ampersand while in message definition mode, it generates a <u>dti</u>(see example in table of op-codes). An ampersand may appear within a message definition if it is coded as an upshift seven. Automatic conversion to ampersand is provided as for colon.

At the present time, a dollar sign($) may not appear at all in the input stream if dictionary change characters also appear within the same problem. The dollar sign is punched by the preprocessor to indicate a dictionary change character. A pound sign(#) may not appear in the input stream if an equal sign also appears. The preprocessor punches a pound sign to indicate an equal sign. See the section on processing preprocessor punch output for further details.

SUGGESTED CODING CONVENTIONS

In view of the fact that further extensions to CW II (via a macro expander) are planned, the labelling scheme shown below will stand the user in good stead when the macro facilities are added.

XXXXXX

- Individual label mnemonic.
  EXAMPLES:
  - XXXXCP for cp label.
  - XXXXC1 for cp c1 label.
  - XXXXEN for end of problem.
- Question identifier.
  EXAMPLES:
  - XXXACP for cp label of question A.
  - XXXA for master label of question A. cp identifiers should be identical to the master label of associated question.
- Problem identifier: A-Z, 0-9.
- Section identifier: A-Z, 0-9.
- Segment identifier: A-Z.

CDC 6600 JOB SUBMISSION

The preprocessor is available on a permanent file at the Computation Center. It may be accessed by the following control card:

READCCF(2973,p_1)

Parameter p_1 allows the user to select one of the three following options:

p_1 = LIST	Generate a source listing; do not process.
p_1 = RUNL	Process and produce a listing of both source and generated statements.
p_1 = RUNLP	Process and produce a listing of both source and generated statements and produce punch output.

A general rule to follow when running jobs on the 6600 is to allow 15(octal) seconds and 20(octal) pages for a problem of 200-250 input cards. Until further notice all options should be run with a field length of 60K words. If expected punch output from any one preprocessor run is greater than 512 cards, an SPL control card(see 6600 User's Manual) must be used to increase the allowed punch limit. The deck structure for a preprocessor run is as follows:

JOB CARD(see 6600 User's Manual)
READCCF(2973,p_1)
7/8/9
↑
USER DECK
↓
EOF

PREPROCESSOR PRINTED OUTPUT

Printed output from the preprocessor necessarily looks quite different from the input card image and some practice will be necessary to become familiar with the notation. A table of preprocessor print characters and their CW II equivalents are shown on the following page.

A line of preprocessor output is broken into three fields. The first field is a flag field of 10 columns and it is in this field that messages to the programmer appear. Other than error numbers of error messages, the message NO PUNCH appears in this field for any statement for which no punch output is generated. The next field is the statement identifier field and consists of the base label of the problem followed by a five-digit octal number. The input card image appears in the third field. The fields are separated by two blanks. An example of preprocessor output is shown below.

```
           FAC00170  FACAEP/
           FAC00200        EPT  16•18¬/¬/12•18¬/¬/¬/FACA≠
           FAC00210        DE   18¬/14≠
           FAC00220        CM   CO≠
           FAC00230        ID   ANSWER PROCESSING≠
           FAC00240        FN   ED¬/R0¬/¬/•D•N¬/¬/ ≠
           FAC00250        LD   ¬/R4≠
           FAC00260        FN   MV¬/R2•40•25¬/R4¬/A≠
           FAC00270        FN   ED¬/R4¬/¬/¬/A≠
           FAC00300  FACAC1≠
           FAC00310        CA   <R¬/>C1≠
NO PUNCH   FAC00310        DM   FAC2≠
           FAC00320        DT   19•00¬/02•19¬/40•00¬/
           FAC00330        HR   FACAEN≠
           FAC00340  FACAW1≠
           FAC00350        WA   <R¬/>W1≠
NO PUNCH   FAC00350        DM   FAC3≠
```

NON-STANDARD PREPROCESSOR

PRINT CHARACTERS

PREPROCESSOR PRINT CHARACTER (p600)	CW TI EQUIVALENT
*	X(TIMES)
[	'(QUOTE)
↓	INDEX
↑	REVERSE INDEX
≡	RETURN
⇄	BACKSPACE
≠	EOB
∧	&(AMPERSAND)
<	SHIFT UP
>	SHIFT DOWN
$	DICTIONARY CHANGE
¬	ALTERNATE CODE
∨	?(QUESTION MARK)

PREPROCESSOR PUNCH OUTPUT

Punch output from the preprocessor will be found in the trays or on the shelves containing punch output in the Computation Center ready room. If more than 512 cards are to be punched during one preprocessor run, the punch limit must be changed by use of the SPL control card(see 6600 User's Manual) before the READCCF control card. The punch output may be interpreted in Hogg Building, room 6.

Each card of preprocessor punch output has a unique identification number appearing in columns 73-80. The sequence consists of the base label of the problem being processed followed by a five-digit octal number. (This card identifier is identical to the statement identifier in the printed output.)

PROCESSING PREPROCESSOR PUNCH OUTPUT

If either an equal sign or a dictionary change character appears within the material which is preprocessed, the punch output must be processed prior to course assembly. The reason for this processing is that the punches corresponding to these characters could not be obtained from 6600 punch output. The equal sign and dictionary change character are punched as a pound sign and dollar sign respectively(hence the restrictions explained in the section on coding conventions). Any card punched on which either of these characters appears is followed by a blank card. The deck must be processed so that the card on which either of these characters appears may be rejected and the corrected statement punched on the following blank card. This processing is accomplished by a program, CTRAN, which is available as a batch job on the IBM 1800 through submission of a job request(see following page) for such processing.in Sutton Hall, room 7(machine room). If neither pound sign nor dollar sign(representing equal sign or dictionary change character) appears in the preprocessor punch output, the deck may be assembled without prior processing by CTRAN.

CAI UTILITY REQUEST

*OPERATIONS STAFF USE ONLY
*( ) job completed, date_____
*( ) reschedule
*( ) partially complete,
reschedule_____

submitted by R. MITCHELL
TODAY'S DATE 4/20/71
telephone no. 471-1044
course name MATHS
segment no. 1
author no. A111

REQUEST INDEX

****

segment registration

( )( ) standard dictionary set 0 1 2 3 (only one set per segment)
( )( ) usable dictionary set(s) 0 1 2 3 (a standard dictionary may also be usable)
( )( ) usable graphic set(s) 1 2 3

****

dictionary/ graphic changes

( )( ) load dictionary set(s) 1 2 3 (circle dictionary set to be loaded)
( )( ) load graphic set(s) 1 2 3 (circle graphic set to be loaded)
( )( ) delete
dictionary set(s) 1 2 3
graphic set(s) 1 2 3
( )( ) equate dictionary set(s) and/or graphic set(s) for
###SPECIFY EQUATION(S) BELOW### course_____

dictionary set(s)		and/or	graphic set(s)	
course	dict. set		course	graphic set
1 to_____	1 2 3		1 to_____	1 2 3
2 to_____	1 2 3		2 to_____	1 2 3
3 to_____	1 2 3		3 to_____	1 2 3

****

macro loading and delete

( )( ) delete macros___ ___ ___ ___ ___ ___ ___ ___
( )( ) load macros ___ ___ ___ ___ ___ ___ ___ ___
( ) entire file

****

course activity

( )( ) delete segment on pack ____)
( )( ) register segment on pack____)
( )( ) card assemble segment on pack____) no. of cards _____
( )( ) insert after label_____sequence number_____
on pack____)
( )( ) list course from pack _____number of listings 1 2 3 4
( ) all the course
( ) partially, from label_____sequence number____
to label_____sequence number____
( )( ) delete user file_____
( )( ) list student file(s)_____

****

miscellaneous jobs

( )( ) duplicate deck_____
( )( ) list deck_____
( )( ) TSX assemble routine_____
( ) card output
( )( ) load function_____

****

OPERATIONS STAFF USE ONLY

job checklist ( ) ALL JOBS AND INFORMATION POSTED

AUDIO REQUESTS.....SEE REVERSE SIDE OF THIS SHEET
SPECIAL REQUESTS....✓.....

audio activity

( )( ) audio address tape as master A AI B BI (circle one)
( )( ) delete symbol table_____
( )( ) assemble symbol table_____(card deck author supplied)
( )( ) list symbol table
( ) list entire table
( ) from symbol_____
to symbol_____
( )( ) audio assembly
narration tape external label_____
type of Master is A or B
and optional address limits
from_____
to_____
( )( ) substitute audio addresses
from symbol table_____

OPERATIONS STAFF USE ONLY

job checklist ( ) ALL JOBS POSTED

SPECIAL REQUESTS

OBJECTIVES:
1. PREPARE DECK FOR ASSEMBLY.
2.
3.
4.
5.

SUGGESTED PROCEDURES

PROCESS WITH CTRAN.

COURSE ASSEMBLY

Assembly of preprocessor punch output, after any necessary processing, is accomplished in the same manner as non-preprocessor material. A request for course assembly(see following page) is submitted with the deck in Sutton Hall, room 7(machine room).

CAI UTILITY REQUEST

*OPERATIONS STAFF USE ONLY
*( ) job completed, date_____
*( ) reschedule
*( ) partially complete, reschedule_____

submitted by R. MITCHELL
TODAY'S DATE 4/20/71
telephone no. 471-1044
course name MATHS
segment no. 1
author no. A111

REQUEST INDEX

segment registration

( )( ) standard dictionary set 0 1 2 3 (only one set per segment)
( )( ) usable dictionary set(s) 0 1 2 3 (a standard dictionary may also be usable)
( )( ) usable graphic set(s) 1 2 3

dictionary/graphic changes

( )( ) load dictionary set(s) 1 2 3 (circle dictionary set to be loaded)
( )( ) load graphic set(s) 1 2 3 (circle graphic set to be loaded)
( )( ) delete
dictionary set(s) 1 2 3
graphic set(s) 1 2 3
( )( ) equate dictionary set(s) and/or graphic set(s) for
###SPECIFY EQUATION(S) BELOW### course_____

dictionary set(s)		and/or	graphic set(s)	
course	dict. set		course	graphic set
1 to_____	1 2 3		1 to_____	1 2 3
2 to_____	1 2 3		2 to_____	1 2 3
3 to_____	1 2 3		3 to_____	1 2 3

macro loading and delete

( )( ) delete macros___ ___ ___ ___ ___ ___ ___ ___
( )( ) load macros ___ ___ ___ ___ ___ ___ ___ ___
( ) entire file

course activity

( )( ) delete segment on pack ____)
( )( ) register segment on pack____)
( )(X) card assemble segment on pack____) No. of cards _____
( )( ) insert after label_____sequence number_____
on pack____)
( )(X) list course from pack ____number of listings (1) 2 3 4
(X) all the course
( ) partially, from label_____sequence number_____
to label_____sequence number_____
( )( ) delete user file_____
( )( ) list student file(s)_____

miscellaneous jobs

( )( ) duplicate deck_____
( )( ) list deck_____
( )( ) TSX assemble routine_____
( ) card output
( )( ) load function_____

OPERATIONS STAFF USE ONLY

job checklist ( ) ALL JOBS AND INFORMATION POSTED

AUDIO REQUESTS.....
SPECIAL REQUESTS.....SEE REVERSE SIDE OF THIS SHEET

UPDATES AND PREPROCESSOR ERRORS

As modifications are made to the preprocessor, updates for the User's Guide will be provided. These updates may be obtained in the CAI Library in room 11 of Sutton Hall or from someone in room 17 of Sutton Hall.

Since the preprocessor is being made available for general use for the first time, it is anticipated that certain undetected errors in the preprocessor will be discovered. If a preprocessor malfunction is discovered by a user, the output from the run along with an attached description of the error, the name of the user, and a telephone number where the user may be contacted should be brought to room 4 of Sutton Hall and left in the mailbox for either Michael Conner or Ronald Mitchell. Questions concerning the preprocessor which are found not to be adequately covered in the User's Guide may be directed to Barbara Sanborn, Michael Conner, or Ronald Mitchell in room 17 of Sutton Hall, telephone 471-1044.

APPENDIX A: KNOWN MALFUNCTIONS

At the present time, only one malfunction is known to exist in the preprocessor and this involves us of the VD op-code.

During variable declaration mode, a buffer specification is made for each variable declared, e.g., 82,05,15, consisting of buffer number, starting position, and number of characters(maximum buffer length). Buffers 2 and 3 are used for loading a definition-set; hence the total number of characters in a definition-set(excluding alternate-coded slashes) must be less than 199, since one position in each buffer must be reserved for EOB.

The situation can arise in which the definition-set will overflow the buffers even though the total number of characters is less than 199. The situation is a result of the order in which the variables are declared since no one variable definition within a definition-set is split between buffers. Consider the following set of declarations:

N1@/72@/75 ≠

N2@/76@/80 ≠

N3@/13@/15 ≠

N4@/07@/10 ≠

N5@/03@/05 ≠

This set of declarations would lead to the following buffer specifications:

VARIABLE	BUFFER SPECIFICATION
N1	B2,00,75
N2	B3,00,80
N3	B3,80,15
N4	OVERFLOW
N5	OVERFLOW

Consider now the following set of declarations:

N1@/72@/75#

N3@/13@/15#

N5@/03@/05#

N2@/76@/80#

N4@/07@/10#

The following set of buffer specifications would be produced:

VARIABLE	BUFFER SPECIFICATION
N1	B2,00,75
N3	B2,75,15
N5	B2,90,05
N2	B3,00,80
N4	B3,80,10

A modification to the preprocessor is in progress which will allow maximum use of buffer storage and will alleviate the situation just described.

APPENDIX B: SAMPLE PROBLEM

On the following pages, a sample problem is traced through the steps from planning guide to preprocessor input to preprocessor output to final course assembly. Special notice should be taken of the manner in which the table is constructed to facilitate keypunching directly from it. The correspondences of the table with the data-set and the pages of the planning guide with the message definitions should also be noted.

The preprocessor input and output listings are annotated to show statements which were entered directly from the planning guide by the keypunch operator and statements generated by the preprocessor. Statements punched directly from the planning guide are labelled KP and generated statements are labelled G. All other statements were coded by the programmer. Special notes also appear on the planning guide.

	η_1	η_2	ζ_1	
1	$\{1, 2\}$	$\{1, 2, 3\}$	yes	look at the member
2	$\{1, 2, 3\}$	$\{1, 2\}$	no	according to you 3
3	$\{1, 2, 3\}$	$\{-1, 0, 1/2, \sqrt{2}\}$	no	according to you 3
4	$\{1, 2, 3\}$	$\{2, 1, 3\}$	yes	look at the member
5	$\{x : x > 0\}$	$\{y : y > 1\}$	no	$1/2 \in \{x : x > 0\}$. Is
6	$\{y : y > 1\}$	$\{x : x > 0\}$	yes	can you think of a
7	$\{x : 0 < x < 1\}$	$\{y : 0 \leq y \leq 1\}$	yes	reconsider your an
8	$\{x : 0 < x < 1\}$	$\{y : 0 < y < 1\}$	yes	reconsider your an
9				
10				
11				
12				
13				
14				
15				
16				
17				

m
1

embers of the sets.

you $3 \in \{1, 2\}$.

you $3 \in \{-1, 0, 1/2, \sqrt{2}\}$.

embers of the sets.

. Is $1/2 > 1$?

of a number that is greater than 1 but not g

ur answer.

ur answer.

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
o	t		g	r	e	a	t	e	r		t	h	a	n		D

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
			C																									
			2																									
		n	o																									
		y	e	s																								
		n	o																									
		y	e	s																								
		y	e	s																								
	?	n	o																									
		n	o																									
		y	e	s																								

5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

TABLE #1

Column

NOTE: EXPANSION IS TO THE RIGHT. NOTE THAT SPACE MUST BE LEFT ON THE DISPLAY LINE FOR VARIABLE TO ASSUME ITS MAXIMUM DISPLAY LENGTH.

Suppose A = ⟨1⟩

and B = ⟨3⟩

Is each member of A also a member of B ?

⟨4⟩

Row (Half line)

Printed in U.S.A.

DAA
(BASE LABEL)

TABLE #1

Column

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

NOTE: EXPANSION IS TO THE RIGHT. NOTE THAT SPACE MUST BE LEFT ON THE DISPLAY LINE FOR VARIABLE TO ASSUME ITS MAXIMUM DISPLAY LENGTH.

pose A = ⟨1⟩

B = ⟨3⟩

each member of A also a member of B ?

⟨4⟩

MASTER LABEL OF QUESTIONA.

False. The answer is $\langle c_1 \rangle$.

Correct.

Is each member of B also a member of A?

$\langle c_2 \rangle$

False. The

r is ⟨c1⟩.

w2 w2

Correct.

c2 c2

member of B also a member of A ?

⟨c3⟩

MASTER LABEL OF QUESTION F.

False.

Carefully consider the question again.

Correct.

False.

refully consider the question again,

wa w1

Correct.

ca c1

```
NO PUNCH   AAA00000          BP  DAA¬/1≠
           DAA00010          VD  1≠
           DAA00020              N1¬/9¬/10≠
           DAA00030              N2¬/16¬/20≠
           DAA00040              C1¬/3¬/5≠
           DAA00050              M1¬/40¬/90≠
           DAA00060              C2¬/3¬/5≠
           DAA00070  ≠
           DAA00100  1¬/¬C1,2¬V¬/¬C1,2,3¬V¬/YES¬/LOOK AT THE MEMBERS OF THE SETS¬/NO≠
           DAA00110  2¬/¬C1,2,3¬V¬/¬C1,2¬V¬/NO¬/ACCORDING TO YOU 3¬E¬C1,2¬V.¬/YES≠
           DAA00120  3¬/¬C1,2,3¬V¬/¬C-1,0,1/2,¬I¬U¬2¬V¬/NO¬/ACCORDING TO YOU 3¬E¬C-1,0,1/2,7X
           DAA00121  2¬V.¬/NO≠
           DAA00130  4¬/¬C1,2,3¬V¬/¬C2,1,3¬V¬/YES¬/LOOK AT THE MEMBERS OF THE SETS.¬/YES≠
           DAA00140  5¬/¬CX<I>X<1>0¬V¬/¬CY<I>Y<1>1¬V¬/NO¬/1/2¬E¬CX<I>X<1>0¬V. <I>S 1/2<1>1¬Ξ
           DAA00141  V¬/YES≠
           DAA00150  6¬/¬CY<I>Y<1>1¬V¬/¬CY<I>X<1>0¬V¬/YES¬/<C>AN YOU THINK OF A NUMBER THAT X
           DAA00151  IS≡↓GREATER THAN 1 BUT NOT GREATER THAN 0V¬/NO≠
           DAA00160  7¬/<C>X<I>0<1>1¬V¬/¬CY<I>0<2*>≥Y<2*>≥1¬V¬/YES¬/RECONSIDER YOUR ANSWER¬Ξ
           DAA00161  .¬/NO≠
           DAA00170  8¬/¬CX<I>0<2>X<2>2¬V¬/¬CY<I>0<1>Y<1>1¬V¬/YES¬/RECONSIDER YOUR ANSWER.¬/X
           DAA00171  YES≠
           DAA00200  ≠
           DAA00210        NO≠
           DAA00220  DAAA≠
KP         DAA00230  3¬/<S>UPPOSE     <A >= ⍕N1⍕≠
           DAA00240  6¬/AND          <B >= ⍕N2⍕≠
           DAA00250  11¬/<I>S EACH MEMBER OF <A >ALSO A MEMBER OF <B >V≠
           DAA00260  DAAAC1≠
           DAA00270  18¬/                  <C>ORRECT.≠
           DAA00300  DAAAW1≠
           DAA00310  18¬/                 <F>ALSE.≠
           DAA00320  21¬/<C>AREFULLY CONSIDER THE QUESTION AGAIN.≠
           DAA00330  24¬/⍕M1⍕≠
           DAA00340  DAAB≠
           DAA00350  20¬/<I>S EACH MEMBER OF <B >ALSO A MEMBER OF <A >V≠
           DAA00360  DAABC2≠
           DAA00370  27¬/                  <C>ORRECT.≠
           DAA00400  DAABW2≠
           DAA00410  24¬/^^^^^^^^^^^^^^^^^^^^^^  <F>ALSE.  <T>HE≠
           DAA00420  27¬/ANSWER IS  ⍕C1⍕ .≠
           DAA00430  ≠
           DAA00440  DAA≠
           DAA00450        PRP≠
```

```
DAA00510  DAAA≠
DAA00520        PR≠
DAA00530        LD   1¬/S2≠
DAA00540        DE   ¬/32≠
DAA00550        DM   DAAA≠
DAA00560  DAAAEP≠
DAA00570        LD   B1¬/B0≠
DAA00600        EP   15,15¬/¬/5,15¬/¬/99¬/DAAA≠
DAA00610        LD   B0¬/B1≠
DAA00620        DE   17¬/14≠
DAA00630        CM   CO≠
DAA00640        FN   ED¬/B0¬/,D¬/¬/ ¬/↓↑¬/↑↓¬/⍟≠
DAA00650        LD   ¬/B4≠
DAA00660        FN   MV¬/:C1:¬/B4¬/A≠
DAA00670        FN   ED¬/B4¬/¬/¬/A≠
DAA00700  DAAAC1≠
DAA00710        CA   B4¬/C1≠
DAA00720        DM   DAAAC1≠
DAA00730        BR   DAAAEN≠
DAA00740  DAAAW1≠
DAA00750        WA   <B¬/>W1≠
DAA00760        LD   0¬/S2≠
DAA00770        DM   DAAAW1≠
DAA01000        BR   DAAAEN≠
DAA01010  DAAAEN≠
DAA01020        CM   EN≠
DAA01030  DAAB≠
DAA01040        PR≠
DAA01050        DE   17¬/14≠
DAA01060        DM   DAAB≠
DAA01070  DAABEP≠
DAA01100        LD   B1¬/B0≠
DAA01110        EPI  24,15¬/¬/5,15¬/¬/99¬/DAAB≠
DAA01120        LD   B0¬/B1≠
DAA01130        DTI  24,23¬/¬/16,23¬/ ≠
DAA01140        DE   26¬/6≠
DAA01150        CM   CO≠
DAA01160        FN   ED¬/B0¬/,D¬/¬/ ¬/↓↑¬/↑↓¬/⍟≠
DAA01170        LD   ¬/B4≠
DAA01200        FN   MV¬/:C2:¬/B4¬/A≠
DAA01210        FN   ED¬/B4¬/¬/¬/A≠
```

```
DAA01220  DAABC2≠
DAA01230          CA   B4¬/C2≠
DAA01240          DM   DAABC2≠
DAA01250          BR   DAABEN≠
DAA01260  DAABW2≠
DAA01270          WA   <8¬/>W2≠
DAA01300          LD   0¬/S2≠
DAA01310          DM   DAABW2≠
DAA01320          BR   DAABEN≠
DAA01330  DAABEN≠
DAA01340          CM   EN≠
DAA01350          AD   1¬/C1≠
DAA01360          CM   LA≠
DAA01370          PT   1¬/DAAA¬/C1≠
DAA01400          ZZ≠
```

```
NO PUNCH    XXX00000          UP  DAA¬/1≠
NO PUNCH    DAA00000          VD  1≠
NO PUNCH    DAA00000              N1¬/9¬/10≠
NO PUNCH    DAA00000              N2¬/16¬/20≠
NO PUNCH    DAA00000              C1¬/3¬/5≠
NO PUNCH    DAA00000              M1¬/40¬/90≠
NO PUNCH    DAA00000              C2¬/3¬/5≠
NO PUNCH    DAA00000  ≠
NO PUNCH    DAA00000  1¬/¬C1,2¬V¬/¬C1,2,3¬V¬/YES¬/LOOK AT THE MEMBERS OF THE SETS¬/NO≠
NO PUNCH    DAA00000  2¬/¬C1,2,3¬V¬/¬C1,2¬V¬/NO¬/ACCORDING TO YOU 3¬E¬C1,2¬V.¬/YES≠
NO PUNCH    DAA00000  3¬/¬C1,2,3¬V¬/¬C-1,0,1/2,¬I¬U*2¬V¬/NO¬/ACCORDING TO YOU 3¬E¬C-1,0,1/2,7X
NO PUNCH    DAA00000  2¬V.¬/NO≠
NO PUNCH    DAA00000  4¬/¬C1,2,3¬V¬/¬C2,1,3¬V¬/YES¬/LOOK AT THE MEMBERS OF THE SETS.¬/YES≠
NO PUNCH    DAA00000  5¬/¬CX<;>X<1>0¬V¬/¬CY<;>Y<1>1¬V¬/NO¬/1/2¬E¬CX<;>X<1>0¬V. <I>S 1/2<1>1¬E
NO PUNCH    DAA00000  V¬/YES≠
NO PUNCH    DAA00000  6¬/¬CY<;>Y<1>1¬V¬/¬CX<;>X<1>0¬V¬/YES¬/<C>AN YOU THINK OF A NUMBER THAT X
NO PUNCH    DAA00000  ISE↓GREATER THAN 1 BUT NOT GREATER THAN 0V¬/NO≠
NO PUNCH    DAA00000  7¬/<C>X<;>0<1>1¬V¬/¬CY<;>0<2*>≥Y<2*>≥1¬V¬/YES¬/RECONSIDER YOUR ANSWER¬E
NO PUNCH    DAA00000  .¬/NO≠
NO PUNCH    DAA00000  8¬/¬CX<;>0<2>X<2>2¬V¬/¬CY<;>0<1>Y<1>1¬V¬/YES¬/RECONSIDER YOUR ANSWER.¬/X
NO PUNCH    DAA00000  YES≠
NO PUNCH    DAA00000  ≠
NO PUNCH    DAA00000          MD≠
NO PUNCH    DAA00000  DAAA≠
NO PUNCH KP DAA00000  3¬/<S>UPPOSE     <A >= :N1:≠
NO PUNCH    DAA00000  6¬/AND          <B >= :N2:≠
NO PUNCH    DAA00000  11¬/<I>S EACH MEMBER OF <A >ALSO A MEMBER OF <B >V≠
NO PUNCH    DAA00000  DAAAC1≠
NO PUNCH    DAA00000  18¬/                    <C>ORRECT.≠
NO PUNCH    DAA00000  DAAAM1≠
NO PUNCH    DAA00000  18¬/                   <F>ALSE.≠
NO PUNCH    DAA00000  21¬/<C>AREFULLY CONSIDER THE QUESTION AGAIN.≠
NO PUNCH    DAA00000  24¬/:M1:≠
NO PUNCH    DAA00000  DAAB≠
NO PUNCH    DAA00000  20¬/<I>S EACH MEMBER OF <B >ALSO A MEMBER OF <A >V≠
NO PUNCH    DAA00000  DAABC2≠
NO PUNCH    DAA00000  27¬/                    <C>ORRECT.≠
NO PUNCH    DAA00000  DAABM2≠
NO PUNCH    DAA00000  24¬/^^^^^^^^^^^^^^^^^^^^^^^^  <F>ALSE.  <I>HE≠
NO PUNCH    DAA00000  27¬/ANSWER IS  :C1: .≠
NO PUNCH    DAA00000  ≠
            DAA00010  DAA≠
            DAA00020          PRB≠
```

```
          DAA00060  CAAA≠
          DAA00070        PR≠
          DAA00100        LU  1¬/S2≠
          DAA00110        DE  ¬/32≠
NO PUNCH  DAA00110        CM  CAAA≠
          DAA00120        LU  <S>UPPOSE     <A >= E↓↓AND          <B >= ¬/B4≠
          DAA00130        FN  MV¬/B2,10,20¬/B4,32¬/IA≠
          DAA00140        FN  MV¬/B2,00,10¬/B4,14¬/IA≠
        G-DAA00150        FN  EU¬/B4¬/¬/¬/^≠
          DAA00160        DT  03,09¬/05,03¬/40,00¬/B4≠
          DAA00170        DT  11,00¬/02,11¬/40,00¬/<I>S EACH MEMBER OF <A >ALSO A MEMBER OF≠
          DAA00171   <B >V≠
          DAA00200  CAAAEP≠
          DAA00210        LU  B1¬/B0≠
          DAA00220        EP  15,15¬/¬/5,15¬/¬/99¬/CAAA≠
          DAA00230        LU  E0¬/B1≠
          DAA00240        DE  17¬/14≠
          DAA00250        CM  CO≠
          DAA00260        FN  EU¬/B0¬/,0¬/¬/ ¬/↑↑¬/↑↑¬/#≠
          DAA00270        LU  ¬/B4≠
          DAA00280        FN  MV¬/B2,30,05¬/B4¬/A≠
          DAA00310        FN  EU¬/B4¬/¬/¬/^≠
          DAA00320  CAAAC1≠
          DAA00330        CA  B4¬/C1≠
NO PUNCH  DAA00330        CM  CAAAC1≠
        G-DAA00340        DT  18,00¬/02,18¬/40,00¬/                <C>ORRECT.≠
          DAA00350        BR  CAAAEN≠
          DAA00360  CAAAW1≠
          DAA00370        WA  <B¬/>W1≠
          DAA00400        LU  C¬/S2≠
NO PUNCH  DAA00400        CM  CAAAN1≠
        G-DAA00410        DT  18,00¬/05,18¬/40,00¬/                <F>ALSE.E↓↓<C>AREFULLY CO≠
          DAA00411  NSIDER THE QUESTION AGAIN.≠
          DAA00420        LU  ¬/B4≠
          DAA00430        FN  MV¬/B3,00,20¬/B4,02¬/IA≠
          DAA00440        FN  EU¬/B4¬/¬/¬/^≠
          DAA00450        DT  24,00¬/02,24¬/40,00¬/B4≠
          DAA00460        BR  CAAAEN≠
          DAA00470  CAAAEN≠
          DAA00500        CM  EN≠
```

```
          DAA00510  CAAB≠
          DAA00520        PR≠
          DAA00530        DE   17¬/14≠
NO PUNCH  DAA00530        DM   CAAB≠
        G DAA00540        DT   20,00¬/02,20¬/40,00¬/<I>S EACH MEMBER OF <R >ALSO A MEMBER OFX
          DAA00541   <A >V≠
          DAA00550  CAABEP≠
          DAA00560        LD   B1¬/B0≠
          DAA00570        EPI  24,15¬/¬/5,15¬/¬/99¬/CAAB≠
          DAA00600        LD   B0¬/B1≠
          DAA00610        DTI  24,23¬/¬/16,23¬/ ≠
          DAA00620        DE   26¬/6≠
          DAA00630        CM   C0≠
          DAA00640        FN   ED¬/B0¬/,D¬/¬/ ¬/**¬/**¬/*≠
          DAA00650        LD   ¬/B4≠
          DAA00660        FN   MV¬/B3,90,05¬/B4¬/A≠
          DAA00670        FN   ED¬/B4¬/¬/¬/A≠
          DAA00700  CAABC2≠
          DAA00710        CA   B4¬/C2≠
NO PUNCH  DAA00710        DM   CAABC2≠
        G DAA00720        DT   27,00¬/02,27¬/40,00¬/                    <C>ORRECT.≠
          DAA00730        BR   CAABEN≠
          DAA00740  CAABW2≠
          DAA00750        WA   <B¬/>W2≠
          DAA00760        LD   C¬/S2≠
NO PUNCH  DAA00760        DM   CAABW2≠
          DAA00770        DTI  24,00¬/02,24¬/19,21¬/XXXXXXXXXXXXXXXXXXXX  <F>ALSE.  <T>HE≠
          DAA01000        LD   ANSWER IS  .¬/B4≠
        G DAA01010        FN   MV¬/B2,30,05¬/B4,10¬/IA≠
          DAA01020        FN   ED¬/B4¬/¬/¬/A≠
          DAA01030        DT   27,00¬/02,27¬/40,00¬/B4≠
          DAA01040        BR   CAABEN≠
          DAA01050  CAABEN≠
          DAA01060        CM   EN≠
          DAA01070        AD   1¬/C1≠
          DAA01100        CM   LA≠
NO PUNCH  DAA01100        PT   1¬/DAAA¬/C1≠
          DAA01110  CAA100≠
          DAA01120        BR   CAA101¬/C1¬/E¬/01≠
        G DAA01130        BR   CAA102¬/C1¬/E¬/02≠
          DAA01140        BR   CAA103¬/C1¬/E¬/03≠
          DAA01150        BR   CAA104¬/C1¬/E¬/04≠
```

```
DAA011..
DAA01170        HR  CAA106-/C1-/E-/06*
DAA01200        HR  CAA107-/C1-/E-/07*
DAA01210        HR  CAA108-/C1-/E-/08*
DAA01220        LD  1-/C1*
DAA01230   CAA101*
DAA01240        LD  -C1,2-V^^^^^-C1,2,3-V^^^^^^^^^^^^^YES^^-/B2*
DAA01250        LD  LOOK AT THE MEMBERS OF THE SETS^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01251   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^NO^^^-/B3*
DAA01260        BR  CAAA*
DAA01270   CAA102*
DAA01300        LD  -C1,2,3-V^^^-C1,2-V^^^^^^^^^^^^^^^^NO^^^-/B2*
DAA01310        LD  ACCORDING TO YOU 3-E-C1,2-V.^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01311   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^YES^^-/B3*
DAA01320        BR  CAAA*
DAA01330   CAA103*
DAA01340        LD  -C1,2,3-V^^^-C-1,0,1/2,-I-UP2-V^^^^^NO^^^-/B2*
DAA01350        LD  ACCORDING TO YOU 3-E-C-1,0,1/2,72-V.^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01351   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^NO^^^-/B3*
DAA01360        HR  CAAA*
DAA01370   CAA104*
DAA01400        LD  -C1,2,3-V^^^-C2,1,3-V^^^^^^^^^^^^^YES^^-/B2*
DAA01410        LD  LOOK AT THE MEMBERS OF THE SETS.^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01411   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^YES^^-/B3*
DAA01420        BR  CAAA*
DAA01430   CAA105*
DAA01440        LD  -CX<I>X<1>0-V^^^-CY<I>Y<1>1-V^^^^^^^^^^^^^^C^^^-/B2*
DAA01450        LD  1/2-E-CX<I>X<1>0-V. <I>S 1/2<1>1V^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01451   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^YES^^-/B3*
DAA01460        HR  CAAA*
DAA01470   CAA106*
DAA01500        LD  -CY<I>Y<1>1-V^^^-CX<I>X<1>0-V^^^^^^^^^^^^^^YES^^-/B2*
DAA01510        LD  <C>AN YOU THINK OF A NUMBER THAT ISE+GREATER THAN 1 BUT NOT GX
DAA01511   REATER THAN 0V^^^^^^^^^^^^^^^^^^NC^^^-/B3*
DAA01520        BR  CAAA*
DAA01530   CAA107*
DAA01540        LD  <C>X<I>J<1>1-V^^^-CY<I>J<2>>2Y<2>>21-V^^^^^^^YES^^-/B2*
DAA01550        LD  RECONSIDER YOUR ANSWER.^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01551   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^NO^^^-/B3*
DAA01560        HR  CAAA*
DAA01570   CAA108*
DAA01600        LD  -CX<I>0<2>X<2>2-V^-CY<I>0<1>Y<1>1-V^^^^^^^^^^^YES^^-/B2*
DAA01610        LD  RECONSIDER YOUR ANSWER.^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^X
DAA01611   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^YES^^-/B3*
DAA016..        HR  CAAA.
```

```
daafe
   1  prr  *e
   2  ld   1/c1*e
   3  ld   0/c2*e
   4  br   daa100*e
daaa*e
   1  dt   0,1/2,0/40,0/DAA*e
   2  pr   *e
   3  ld   1/s2*e
   4  de   0/32*e
   5  ld   Suppose    A = *c*i*iand        B = /b4*e
   6  fn   mv/b2,10,20/b4,32/ia*e
   7  fn   mv/b2,00,10/b4,14/ia*e
   8  fn   ed/b4///&*e
   9  dt   3,0/5,3/40,0/b4*e
  10  dt   11,0/2,11/40,0/Is each member of A also a member of B ?*e
daaaep*e
   1  ld   b1/b0*e
   2  ep   15,15/2,15/5,15//99/daaa*e
   3  ld   b0/b1*e
   4  de   17/1 *e
   5  no   co*e
   6  no   control logic*e
   7  no   em*e
   8  fn   ed/b0/,d// /*i*r/*r*i/*b*e
   9  ld   /b4*e
  10  fn   mv/b2,30,05/b4/a*e
  11  fn   ed/b4///&*e
daaacl*e
   1  ca   b4/c1*e
   2  dt   18,0/2,18/40,0/                Correct.*e
   3  br   daaaen*e
daaawl*e
   1  wa   */w1*e
   2  ld   0/s2*e
   3  dt   18,0/5,18/40,0/               False.*c*i*iCarefully consider the quest
ion again,*e
   4  ld    /b4*e
   5  fn   mv/b3,0,90/b4,0/ia*e
   6  fn   ed/b4///&*e
   7  dt   24,0/2,24/40,0/b4*e
```

```
daaaen*e
   1  no  en*e
   2  pr  *e
   3  dti 28,39⁄2,28⁄1,39⁄S*e
   4  ld  b1⁄b0*e
   5  epi 30,1⁄2,30⁄1,39⁄⁄1⁄zz*e
   6  ld  b0⁄b1*e
   7  pr  *e
   8  no  em*e
daab*e
   1  pr  *e
   2  de  17⁄14*e
   3  dt  20,0⁄2,20⁄40,0⁄Is each member of B also a member of A ?*e
daabep*e
   1  ld  b1⁄b0*e
   2  epi 24,15⁄2,24⁄5,15⁄⁄99⁄daab*e
   3  ld  b0⁄b1*e
   4  dti 24,23⁄2,24⁄16,23⁄ *e
   5  de  26⁄6*e
   6  no  co*e
   7  no  control logic*e
   8  no  em*e
   9  fn  ed⁄b0⁄,d⁄⁄ ⁄*i*r⁄*r*i⁄*b*e
  10  ld  ⁄b4*e
  11  fn  mv⁄b3,90,05⁄b4⁄a*e
  12  fn  ed⁄b4⁄⁄⁄&*e
daabc2*e
   1  ca  b4⁄c2*e
   2  dt  27,0⁄2,27⁄40,0⁄                    Correct.*e
   3  br  daaben*e
daabw2*e
   1  wa  *⁄w2*e
   2  ld  0⁄s2*e
   3  dti 24,0⁄2,24⁄10,21⁄xxxxxxxxxxxxxxxxxxxx  False.  The*e
   4  ld  answer is  .⁄b4*e
   5  fn  mv⁄b3,90,05⁄b4,10⁄ia*e
   6  fn  ed⁄b4⁄⁄⁄&*e
   7  dt  27,0⁄2,27⁄40,0⁄b4*e
   8  br  daaben*e
```

```
daaben*e
   1  pr   *e
   2  dti  28,39⁄2,28⁄1,39⁄S*e
   3  ld   b1⁄b0*e
   4  epi  30,1⁄2,30⁄1,39⁄⁄1⁄zz*e
   5  ld   h0⁄h1*e
   6  pr   *e
   7  no   em*e
   8  ad   1⁄c1*e
   9  no   la*e
  10  br   #01#56⁄s2⁄0*e
  11  ad   1⁄c2*e
  12  br   pr1⁄c2⁄g⁄3*e
#01#56*e
   1  no   em*e
daal00*e
   1  br   daal01⁄c1⁄e⁄1*c
   2  br   daal02⁄c1⁄e⁄2*c
   3  br   daal03⁄c1⁄e⁄3*e
   4  br   daal04⁄c1⁄e⁄4*c
   5  br   daal05⁄c1⁄e⁄5*e
   6  br   daal06⁄c1⁄e⁄6*e
   7  br   daal07⁄c1⁄e⁄7*e
   8  br   daal08⁄c1⁄e⁄8*e
   9  ld   1⁄c1*c
daal01*e
   1  ld   ⊳1,2⊲&&&&⊳1,2,3⊲&&&&&&&&&&&&&&yes&&⁄b2*e
   2  ld   look at the members of the sets&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&no&&&⁄b3*c
   3  br   daaa*e
daal02*e
   1  ld   ⊳1,2,3⊲&&&&⊳1,2⊲&&&&&&&&&&&&&&&&&no&&&⁄b2*c
   2  ld   according to you 3⊳⊳1,2⊲.&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&yes&&⁄b3*c
   3  br   daaa*e
daal03*c
   1  ld   ⊳1,2,3⊲&&&&⊳-1,0,1/2,ʃa*b?⊲&&&&&no&&&⁄b2*c
   2  ld   according to you 3⊳⊳-1,0,1/2,7?⊲.&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&no&&&⁄b3*c
   3  br   daaa*c
```

```
daal04*c
   1  ld  ⊳1,2,3⊲&&&⊳2,1,3⊲&&&&&&&&&&&&&yes&&⁄b2*c
   2  ld  look at the members of the sets.&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&yes&&⁄b3*c
   3  br  daaa*c
daal05*c
   1  ld  ⊳x;x>0⊲&&&⊳y;y>1⊲&&&&&&&&&&&&&no&&⁄b2*c
   2  ld  1/2⊳⊳x;x>0⊲. Is 1/2>1?&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&yes&&⁄b3*c
   3  br  daaa*c
daal06*e
   1  ld  ⊳y;y>1⊲&&&⊳x;x>0⊲&&&&&&&&&&&&&yes&&⁄b2*e
   2  ld  Can you think of a number that is*c*igreater than 1 but not greater th
an 0?&&&&&&&&&&&&&&&&no&&&⁄b3*e
   3  br  daaa*e
daal07*c
   1  ld  Cx;0>1⊲&&&⊳y;0<*b_y<*b_1⊲&&&&&&&yes&&⁄b2*c
   2  ld  reconsider your answer.&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&no&&&⁄b3*c
   3  br  daaa*c
daal08*e
   1  ld  ⊳x;0<x<2⊲&⊳y;0>y>1⊲&&&&&&&&&&&yes&&⁄b2*e
   2  ld  reconsider your answer.&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&yes&&⁄b3*c
   3  br  daaa*c
   4  en  *e
```

GLOSSARY OF TERMS

Base-label	Three-character label used to identify a problem. These characters are used as the first three characters of all labels generated by the preprocessor during problem processing.
Data-set	A series of definition-sets (see below) from which a table of variable values is created.
Definition-set	A set of values(definitions) to be taken by declared variables.
Expansion direction	The direction, from the variable reference, in which the literal string value of the variable is to be extended during substitution for the variable. If right expansion, replacement begins with leftmost colon; if left expansion, replacement begins with rightmost colon. (See sample problem in App. B for examples.)
Maximum buffer length	Within the data-set, the maximum number of buffer positions(including positions for control characters) needed to set up the literal string representation of the variable for display.
Maximum display length	Within the data-set, the maximum length(number of characters) assumed by the variable.
Maximum negative deviation	Any literal string value taken by the variable is assumed to require 2 display rows(one line) on the CRT. The maximum negative deviation is the maximum number of rows(half-lines) <u>below</u> this basic display line required to display the string.
Maximum positive deviation	Same as maximum negative deviation except the measure is of the number of rows <u>above</u> the basic display line required.

Message body	The series of statements of the type (ROW No.)0/TEXT which follow a message name and define that message.
Message content data	Display templates (pages of a planning guide) which provide a pattern for presentation of variable data.
Variable data	Variable length literal strings which may take on new values for repeated use with a display template.